



Organización
de las Naciones Unidas
para la Educación,
la Ciencia y la Cultura



Cátedra UNESCO
Tecnologías de apoyo para
la Inclusión Educativa



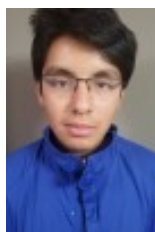
REVISTA

JUVENTUD Y CIENCIA SOLIDARIA:

En el camino de la investigación

PROTOTIPO DE ASISTENTE VIRTUAL EN PYTHON

Diego Patricio Rivera Guachichulca



Diego Patricio Rivera Guachichulca, nació el 4 de diciembre del 2004, tengo 16 años. Soy estudiante de la Unidad Educativa Técnico Salesiano, actualmente pertenezco al grupo del 2E2. Soy un apasionado por el desarrollo web, me gustan los videojuegos y siempre me ha interesado el mundo de la programación.

Resumen

El siguiente artículo trata sobre un asistente virtual para personas con discapacidad y para el aumento de efectividad en el trabajo.

El primer prototipo tenía varios fallos y solo abría páginas web y algunas aplicaciones, la cantidad era muy limitada.

Además de que no solía reconocer muy bien los comandos y las instrucciones.

Teniendo eso en cuenta presento la mejora de ese asistente con la diferencia de que este tiene una estructura superior y la implementación de funciones para que

tenga un rendimiento más aceptable, el mismo está realizado en el lenguaje de programación Python, tiene varias bibliotecas y módulos que facilitaron su desarrollo y ayudan a que sea más eficiente.

Dichas bibliotecas y módulos son de libre acceso lo que hace que cualquier persona pueda utilizarlas, además de que tienen documentación en la página oficial de Python lo que ayudará a tener una idea de cómo funciona cada uno. En dichas documentaciones hay un ejemplo para asegurar el buen uso del módulo o la biblioteca.

El objetivo de este asistente es el mismo que el anterior y también fue desarrollado en un IDE de Microsoft (Visual Studio Code) que es muy similar al entorno de desarrollo Visual Studio utilizado para el anterior asistente. La razón para cambiar a Visual Studio Code es porque al trabajar con Python en ese IDE es muy cómodo lo que facilita el desarrollo del asistente.

Palabras clave: IDE, Python, función, PIP, CMD

Explicación del tema

El lenguaje de programación Python es considerado uno de los mejores y utilizado en varios campos: desarrollo web, machine learning, análisis de datos, programación orientada a objetos y muchos más.

Pero en esta ocasión se emplea para realizar un asistente virtual junto al CMD para instalar las bibliotecas; los módulos no hay necesidad de instalarlos ya que vienen junto a Python.

El proceso es abrir el CMD

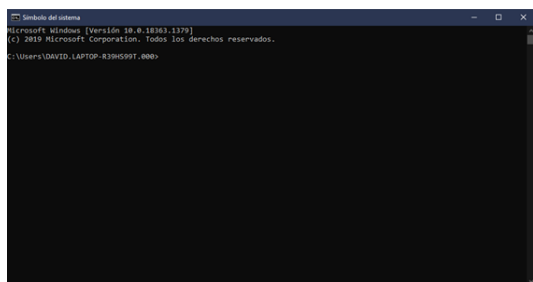


Figura 1. CMD de Windows

Fuente: Autor

Y buscar la biblioteca en la página de Python para que nos dé el PIP.

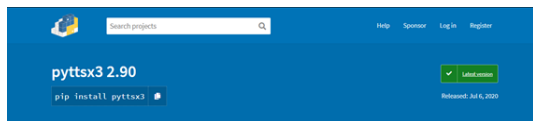


Figura 2. Página de Python para obtener el instalador de la Biblioteca

Fuente: Autor

Eso pegamos en el CMD para que se nos instale la Biblioteca.

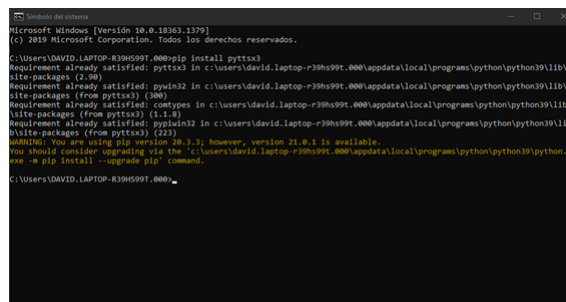


Figura 3. CMD de Windows después de instalar una biblioteca

Fuente: Autor

Lo siguiente es instalar los módulos y bibliotecas necesarias; es de recordar que las bibliotecas se instalan aparte y están marcadas en la imagen con un #pip.

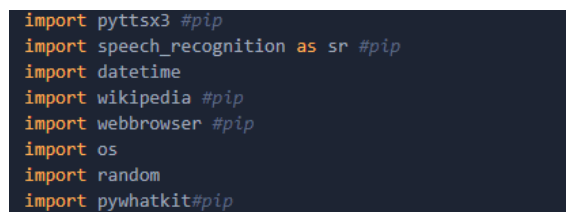


Figura 4. IDE Visual Studio Code, código con las bibliotecas necesarias

Fuente: Autor

Luego se definen todas las variables que se va a usar, además de eso se habilita el parlante para que el asistente responda con la voz predeterminada de Windows.

Eso definirá el booleano para iniciar el asistente y mantenerlo activo.

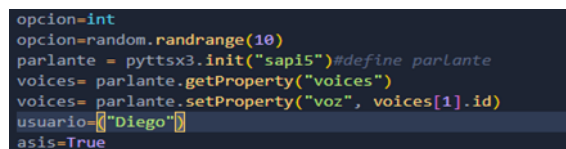


Figura 5. Variables del código

Fuente: Autor

Ya se tiene los modelos, bibliotecas y variables necesarias del asistente, lo siguiente será definir las funciones para acortar líneas de código, para que se vuelva más eficiente.

En este caso se usan tres funciones; una para que el asistente nos responda, otra para que nos salude cada vez que lo activemos y la tercera que es la más importante será para recibir órdenes.

Primera función

Esta será la función para que el asistente pueda hablar, se debe definir la función.

```
def hablar (text):
```

Figura 6. Se define la primera función
Fuente: Autor

Y según la tabulación se tiene que poner los siguientes atributos a la variable de parlante para que funcione.

```
parlante.say(text)
parlante.runAndWait()
```

Figura 7. Parlante ya funcionando
Fuente: Autor

Cada vez que se llame a esta función, el parlante se reproducirá con la voz predeterminada de Windows el texto recibido.

Segunda función

Además de lo ya hecho anteriormente, se establece un valor a la variable hora, con la instrucción `datetime.datetime.now().hour`. Python tomará la hora y le pondrá ese valor, luego se pondrá que imprima la hora en pantalla para comprobar que sirve.

```
def Buenas():
    hora=int(datetime.datetime.now().hour)
    print(hora)
```

Figura 8. Segunda función
Fuente: Autor

```
23
```

Figura 9. Hora que nos da el asistente
Fuente: Autor

```
23:27
```

Figura 10. Hora del computador
Fuente: Autor

A continuación, se definen condicionales para que dependiendo de la hora el asistente salude por el nombre dado.

```
if hora>=0 and hora<12:
    hablar("Buenos días"+ usuario+"...")
elif hora>=12 and hora<18:
    hablar("Buenas tardes"+ usuario+"...")
else:
    hablar("Buenas noches"+ usuario+"...")
```

Figura 11. Saludo con usuario
Fuente: Autor

Tercera función

Además de definir la función, también se define que tome el audio del micrófono y pueda tomar las instrucciones dadas; se pide que imprima en pantalla para asegurar que esté recibiendo órdenes y probar que está funcionando.

```
def recibir():
    r=sr.Recognizer()
    with sr.Microphone() as recurso:
        print("Escuchando instrucciones")
        audio=r.listen(recurso)
```

Figura 12. Tercera función con el micrófono definido
Fuente: Autor

Después de esto usaremos un *try* para asegurar que el asistente reciba las órdenes; si el asistente no logra reconocer ninguna instrucción se lo tomará como error y lo notificará, esto se realiza para garantizar que su funcionamiento sea correcto. En caso de que la instrucción se entienda retornará lo que entendió.

```
try:
    print("Reconociendo instruccion")
    ins=r.recognize_google(audio)
    print("Se entendio ",ins)
    asis==True
except Exception:
    print("No se entendio ninguna instruccion")
    ins=None
return ins
```

Figura 13. Try para que asegurar que se reciba instrucciones
Fuente: Autor

```
Iniciando asistente
0
Escuchando instrucciones
Escuchando instrucciones
Reconociendo instruccion
No se entendio ninguna instruccion
```

Figura 14. Asistente no reconoció ninguna instrucción
Fuente: Autor

Estructura principal del asistente

Después de definir todas las funciones se empieza con el asistente, para programar que salude; además de eso, se usará el booleano en True para indicar que se quiere que se mantenga activo.

```
hablar ("Iniciando asistente...")
Buenas()
while asis==True:
    #poner aquí while para mantener el asistente activo
    print("Escuchando instrucciones")
    ins = recibir()
```

Figura 15. Inicio de la estructura del asistente
Fuente: Autor

Después de esto se hará que el asistente sea capaz de buscar personas en Wikipedia usando la biblioteca.

Para esto se comprueba que la palabra Wikipedia esté en la instrucción.

```
if "wikipedia" in ins.lower():
```

Figura 16. Comprobación para buscar datos en Wikipedia
Fuente: Autor

Después de esto se escribe una línea para que se escuche el personaje que queremos buscar, se lo obtiene de Wikipedia con la instrucción wikipedia.summary, se le dirá que diga el resultado y lo imprima en pantalla.

```
if "wikipedia" in ins.lower():
    hablar("Que desea buscar en wikipedia...")
    ins=ins.replace("wikipedia","")
    resultado= wikipedia.summary(ins,sentences=2)
    print(resultado)
    hablar(resultado)
```

Figura 17. Búsqueda en Wikipedia
Fuente: Autor

Como la biblioteca de Wikipedia está en inglés el resultado se obtendrá en ese idioma.

```
Escuchando instrucciones
Reconociendo instrucción
Se entendió wikipedia Diego Rivera
Diego Paría de la Concepción Juan Nepomuceno Estanislao de la Rivera y Barrientos Acosta y Rodríguez, known as Diego Rivera (Spanish pronunciation: [ˈdjeɣo ɾiˈβeɾa]; December 8, 1886 – November 24, 1957), was a prominent Mexican painter. His large frescoes helped establish the mural movement in Mexican and international art.
```

Figura 18. Resultado Búsqueda
Fuente: Autor

Después se crea una instrucción para que diga la hora exacta; se realiza lo mismo que la función de saludo, pero con la diferencia de que se va a conseguir los datos de las horas, minutos y segundos con la orden tiempo=datetime.datetime.now().strftime("%H:%M:%S")

```
elif "hora" in ins.lower():
    tiempo=datetime.datetime.now().strftime("%H:%M:%S")
    hablar(usuario+"la hora es"+tiempo+"...")
```

Figura 19. Hora
Fuente: Autor

```
Escuchando instrucciones
Reconociendo instrucción
Se entendió que hora es
Diegola hora es00:22:27...
```

Figura 20. Asistente da la hora
Fuente: Autor

Después se programa que recomiende música, el asistente podrá abrir la canción en YouTube, para esto

se utilizará la biblioteca webbrowser y la instrucción webbrowser, se usa una variable cuyo valor generará números al azar, dependiendo abrirá una pestaña en YouTube con la canción que corresponda a la opción que salió.

```
elif "musica" in ins.lower():
    print(opcion)
    if opcion==0:
        webbrowser.open("https://www.youtube.com/watch?v=Pz1VjgKzy8Y")
        hablar("Que te parece esta sugerencia"+usuario+"...")
    elif opcion==1:
        webbrowser.open("https://www.youtube.com/watch?v=hUq6KkZcUY")
        hablar("Que te parece esta sugerencia"+usuario+"...")
    elif opcion==2:
        webbrowser.open("https://www.youtube.com/watch?v=x4-ZD1mvzUI")
        hablar("Que te parece esta sugerencia"+usuario+"...")
    elif opcion==3:
        webbrowser.open("https://www.youtube.com/watch?v=k8U8gJfX9Vvk")
        hablar("Que te parece esta sugerencia"+usuario+"...")
    elif opcion==4:
        webbrowser.open("https://www.youtube.com/watch?v=k8U8gJfX9Vvk")
        hablar("Que te parece esta sugerencia"+usuario+"...")
    elif opcion==5:
        webbrowser.open("https://www.youtube.com/watch?v=ixx0JuV-T_E")
        hablar("Que te parece esta sugerencia"+usuario+"...")
    elif opcion==6:
        webbrowser.open("https://www.youtube.com/watch?v=nZfmaFB5cgY")
        hablar("Que te parece esta sugerencia"+usuario+"...")
    elif opcion==7:
        webbrowser.open("https://www.youtube.com/watch?v=CIZ5S2cgh0Y")
        hablar("Que te parece esta sugerencia"+usuario+"...")
    elif opcion==8:
        webbrowser.open("https://www.youtube.com/watch?v=HkbG39-T4H0")
```

Figura 21. Recomendación de música
Fuente: Autor

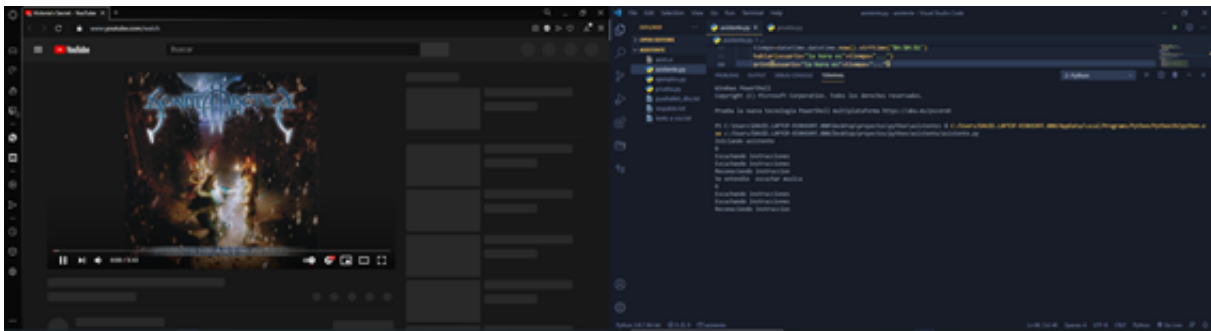


Figura 22. Asistente abre YouTube con la canción obtenida en la opción 6
Fuente: Autor

Al igual que se pide abrir YouTube con una canción Esemtia, para esto es necesario copiar el enlace de la se puede hacer con cualquier página web, por ejemplo, página y ponerlo en la instrucción webbrowser.

```
elif "esemtia" in ins.lower():
    webbrowser.open("https://edu.esemtia.ec/LoginEsemtia.aspx")
    hablar("Abriendo esemtia...")
```

Figura 23. Esemtia en la instrucción webbrowser
Fuente: Autor

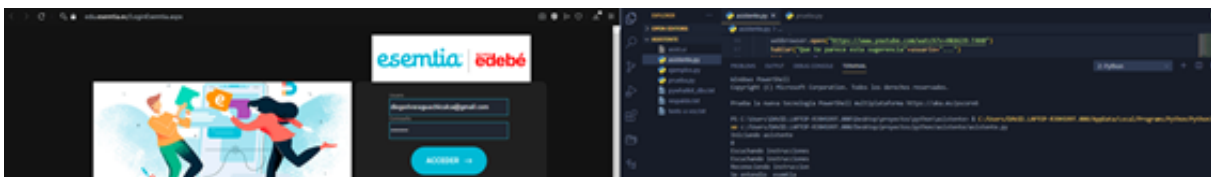


Figura 24. Asistente abre Esemtia
Fuente: Autor

```
elif "soporte" in ins.lower() :
    os.startfile("C:\\Users\\DAVID.LAPTOP-R39HS99T.000\\Desktop\\Soporte Técnico")
    hablar("Abriendo carpeta de soporte tecnico...")
```

Figura 25. Asistente abre Esemtia
Fuente: Autor

Pero no solo puede hacer eso, sino que también mos es usar el módulo os y la ruta, la instrucción puede abrir cualquier archivo, lo único que necesita- necesaria es os.startfile.

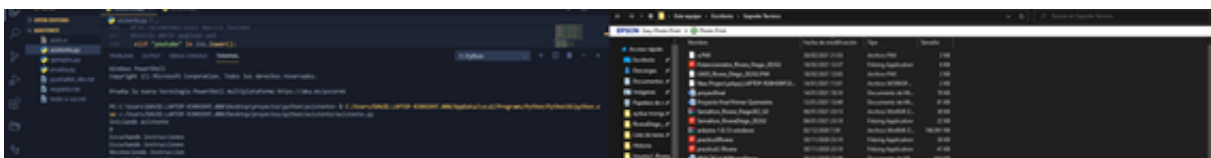


Figura 26. Asistente debe abrir la carpeta de soporte técnico
Fuente: Autor

Conclusiones

Este asistente puede cumplir su objetivo principal, pero aún le falta mucho, por ejemplo, abrir aplicaciones y una interfaz gráfica para que sea más fácil usarlo; hasta el momento su desarrollo muestra superioridad al asistente anterior ya que se tarda menos en recibir órdenes, es más preciso, y cuenta con instrucciones extra, además de que a diferencia del anterior este necesita reconocer una palabra en vez de una oración completa y tampoco requiere guardar las palabras que se desean capturar, mientras que el otro asistente si lo necesitaba, y eso reducía considerablemente su eficacia y rendimiento.

Se espera que en un futuro cuando la estructura principal del asistente esté terminada se le pueda implementar machine learning (inteligencia artificial) para que no sea necesario reconocer la misma palabra clave, sino que vaya aprendiendo por sí solo todas las pala-

bras para pedir una instrucción y las diferencie para que su eficacia sea mucho mayor a la de la actual.

Referencias

- [1] Python, Instalation, 2020. Obtenido de shorturl.at/kpFN3
- [2] Python, SpeechRecognition 3.8.1., 2020. Obtenido de shorturl.at/chwxV
- [3] Python, Wikipedia, 2020. Obtenido de shorturl.at/yBEO3
- [4] Python, pycopy-webbrowser 0.0.0, 2020. Obtenido de shorturl.at/hjAL3
- [5] Python, pywhatkit 3.9, 2020. Obtenido de shorturl.at/bfASU
- [6] Python, Miscellaneous operating system interfaces, 2020. Obtenido de shorturl.at/bqPY0
- [7] Python, Basic date and time types, 2020. Obtenido de shorturl.at/gnyQ2